Noise Robust Online Inference for Linear Dynamic Systems

Saikat Saha

Abstract

We revisit the Bayesian online inference problems for the linear dynamic systems (LDS) under non-Gaussian environment. The noises can naturally be non-Gaussian (skewed and/or heavy tailed) or to accommodate spurious observations, noises can be modeled as heavy tailed. However, at the cost of such noise robustness, the performance may degrade when such spurious observations are absent. Therefore, any inference engine should not only be robust to noise outlier, but also be adaptive to potentially unknown and time varying noise parameters; yet it should be scalable and easy to implement.

To address them, we envisage here a new noise adaptive Rao-Blackwellized particle filter (RBPF), by leveraging a hierarchically Gaussian model as a proxy for any non-Gaussian (process or measurement) noise density. This leads to a conditionally linear Gaussian model (CLGM), that is tractable. However, this framework requires a valid transition kernel for the intractable state, targeted by the particle filter (PF). This is typically unknown. We outline how such kernel can be constructed provably, at least for certain classes encompassing many commonly occurring non-Gaussian noises, using auxiliary latent variable approach. The efficacy of this RBPF algorithm is demonstrated through numerical studies.

Index Terms

noise adaptive filter; Rao-Blackwellized particle filter; noise robust inference; Kalman filter; asymmetric noise;

I. Introduction

Many applications of interest to the signal processing community (e.g., tracking, autonomous navigation and surveillance) require that the inference must be performed on near real-time (online) using the

S. Saha is with the Division of Automatic Control, Department of Electrical Engineering, Linköping University, Sweden, saha@isy.liu.se

streaming sensor data. The stability and the performance of the underlying inference engines however depend crucially on the sensor data quality. Usually, any spurious sensor data needs to be detected and discarded before passing to the inference engine, so that the latter is not susceptible to permanent failure. In recent times, the sensors are becoming cheaper, however at the expense of their performance reliability. The proliferation of such inexpensive sensors has opened up the possibility to explore many complex and high dimensional problems (e.g., motion tracking, road traffic monitoring), which are hitherto impossible with a limited number of costly sensors. This trend for using inexpensive sensors has in turn, laid greater emphasis on the processing algorithms to the extent that any inference algorithm (presumably with more computing prowess) is required to be robust and stable against such spurious sensor data; yet simple to implement and vastly scalable to the high dimensional problems. Against this backdrop, we consider the online inference problems for the discrete time LDS.

A. Problem Background

Consider the following discrete time LDS relating the latent state $x_k \in \mathbb{R}^{n_x}$ at time step k to the observation $y_k \in \mathbb{R}^{n_y}$ as

$$x_k = A_k \; x_{k-1} + B_k \; w_k, \tag{1a}$$

$$y_k = C_k \; x_k + e_k, \tag{1b}$$

where w_k and e_k are the process and measurement noise respectively. The noises are assumed to be independent and also independent of each other. The model parameters $\{A_k, B_k, C_k\}$ are considered to be known here. Given this model, an initial state prior (i.e., $p(x_0)$) and a stream of observations up to time k, $y_{0:k} \triangleq \{y_0, y_1, \ldots, y_k\}$, one typical inference task is to optimally estimate the sequence of (posterior) densities $p(x_k|y_{1:k})$, in an online fashion over time. This is known as the online Bayesian filtering problem for the LDS, and the density $p(x_k|y_{1:k})$ is called as the filtering density. When the noises above are Gaussian, the filter density can be obtained recursively in closed form using the celebrated Kalman filter (KF) [1]–[3]. However, this analytical tractability is lost if any noise deviates from such nominal Gaussian assumptions.

In reality, many noise sources naturally appear to be non-Gaussian (characterized by their heavy tails and/or skewness). For example, noise with an impulsive nature (sharp spikes and/or occasional bursts) appears in many applications such as speech and audio signal processing, astrophysical imaging, underwater navigations, multi-user radar communications, kick detection in oil drilling, finance and insurance among others [4]–[11]. This impulsive nature can be modeled e.g., by a noise distribution that has heavier tails than the Gaussian distribution. Heavy tailed distributions are also used to model the presence of so called *outliers*, which are data points, that do not appear to follow the pattern of the other data points [12]. Data from the visual sensors, GPS devices, sonar, and radar are often contaminated by such outliers. The root causes of such outlying observations are often unknown or are excluded deliberately from the model due to the complexity and the computational issues. So under such simplified modeling of complex real world processes, these unusual observations can be taken care by noise outliers. This in turn, requires heavy tailed distributions for the noises [13].

In the filtering context, when a nominal model with specified Gaussian noises cannot account for the outliers or sudden change in unknown input signals, the filter becomes unstable. Since for the real world applications, often we have weak knowledge on the systems and outliers are frequent, naturally, the filtering problems under heavy tailed noises have attracted considerable research attentions [14]–[17]. We note that in the context of filtering, the process and/or the observation noise can have heavy tails. Although heavy tailed observation noise have received much attention, in applications like maneuvering target tracking, modeling occasional pilot induced changes requires a heavy tailed process noise [18]. In contrast, although skewness appears in many applications [19], the associated online inference problem has not been well explored [1].

Non-Gaussian noise can also appear due to the modeling artifact. We illustrate the last point through the following example:

1) Stochastic volatility model: We consider the following discrete time model [21]

$$h_{k+1} = \gamma_0 + \gamma_1 h_k + \eta_k, \tag{2a}$$

$$y_k = \epsilon_k exp(h_k/2), \tag{2b}$$

where h_k and y_k are the latent log-volatility and observed asset return at time step k with $|\gamma_1| < 1$, $h_0 \sim \mathcal{N}(0, \frac{\sigma_n^2}{1-\gamma_1^2})$, $\eta_k \overset{iid}{\sim} \mathcal{N}(0, \sigma_n^2)$ and $\epsilon_k \overset{iid}{\sim} \mathcal{N}(0,1)$, where *iid* means independent and identically distributed. γ_0, γ_1 and σ_n are assumed to be known. Here η_k is uncorrelated to ϵ_k, i.e. no leverage effect is considered. Note that the measurement noise is multiplicative. The above dynamic model can equivalently be cast as a linear state space model:

$$h_{k+1} = \gamma_0 + \gamma_1 h_k + \eta_k, \tag{3a}$$

$$log(y_k^2) = h_k + log(\epsilon_k^2). \tag{3b}$$

[1] One notable exception is the recent article [20], that has been brought to our notice.

However the observation noise $e_k \equiv log(\epsilon_k^2)$, being log of a χ^2 random variable, is no more Gaussian. In fact, the density of this noise is available analytically, given by (see e.g., [22])

$$f(e_k) = \frac{1}{\sqrt{2\pi}} exp\big\{ - 0.5\big(exp(e_k) - e_k\big)\big\}, \ -\infty < e_k < \infty. \tag{4}$$

From Figure (1), it is evident that the density is highly skewed. Thus, we have a LDS driven by a non-Gaussian (measurement) noise.

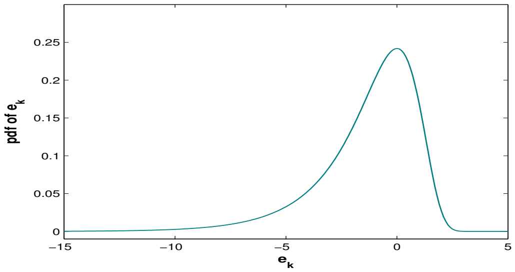


Figure 1: Density plot for the observation noise in (4).

While the noise distribution is certainly a key factor for the inference task, the other equally important aspect for practical considerations is the knowledge on the noise statistics. In many practical contexts, although prior knowledge on the noise distributions is fairly available, the corresponding noise parameters are often unknown. Thus additionally, the noise parameters need to be estimated on the fly as well. This is known as online noise adaptive filtering. When the noise parameters are stationary, the online Bayesian estimation of the parameters is known to be difficult and one practical solution is to assume the noise parameters to be slowly varying in time [23]. However, this assumption easily breaks down in the presence of any potential outlier. To get rid of the outliers, some mechanisms are usually placed to detect and immediately discard them. However such outlying observations may carry important information about e.g., any unmodeled system characteristics and model degradation and as such, as pointed out in [24], ”Routinely ignoring unusual observations is neither wise nor statistically sound”. On the other hand, eventhough such outliers can be accommodated by using e.g., a properly specified heavy tailed noise distribution from a known parametric family, the performance of the filter can be degraded severely when the outliers are absent. This happens due to the use of fixed form of distribution. To alleviate this problem in noise adaptive filtering, essentially what we need is to use a heavy tailed noise, whose parameters are time varying. However, coping with nonstationary noise parameters in online setting is very challenging as the corresponding dynamics for the parameters cannot be easily specified *a priori* in practice.

B. Prior works

There is a long history on the efforts of improving the KF algorithm under the non-Gaussian environments. The earlier attempts were mainly based on the robustification arguments in the presence of outliers. These were primarily addressed either by analytical approximations using elliptical noise distributions or by heuristic cost functions in the update step of KF. The elliptical noise based approaches [25], [26] were not robust, as the posterior mean is unbounded as a function of the residuals, whereas the approaches based on the ad-hoc cost functions (e.g., [27], [28]) requires tuning of parameters and their implementations were very involved as compared to KF. Later, the simulation based approaches (e.g., PF) [24], [29] became popular due to their ease of implementations. However, the major disadvantages were their non-scalability together with computational costs [30]. Several recent articles have addressed the filtering problem using variational Bayes (VB) [14], [15], [17] and convex optimization based methods [31]. Although VB method is quite scalable, in general, it requires fairly involved mathematical derivations and is known to consistently underestimate the posterior covariance. Common to all these recent studies is the assumption of outliers in the observation noise only. The methods by [15], [17], [31] do not accommodate any persistence (time correlated) noise outliers. Moreover, [15] considers a heuristic transition model for the noise parameter (as observed by [14], the filter is not stable under abrupt change in noise) whereas [14], [31] require additional input parameter from the user. More recently, observing that the VB based algorithms are very sensitive to process noise outlier, [16] proposed an analytical approximation based t filter, where both the process and observation noises are Students' t. The approximations here require that the estimated state and process/observation noise is jointly t with a common degree of freedom (dof) parameter. To maintain these hypotheses and to prevent the growth of dof parameters in filter recursion, again careful tuning is required at each step. For all the cases, the noise class is implicitly assumed to be symmetric and Students' t.

From the literature above, we see that many of the existing approaches require manual tuning of the parameters and also their implementations are substantially involved as compared to well understood KF. Thus any future inference framework should ideally be free of such heuristic tuning and the implementation should be simple to understand. Also, any future framework should be able to handle-(a) asymmetric noise and (b) symmetric heavy tailed noise other than Students' t.

C. *Sketch of our contributions*

Keeping in view the points above, we consider the online inference for the LDS under a fairly general and realistic scenario. This is addressed here in two stages by successively dropping the following common assumptions: (a) noises are Gaussian and (b) noise parameters are known.

In the first stage, we consider the LDS under (known) stationary non-Gaussian environment. The corresponding inference task is analytically intractable. In our approach, we envisage a hierarchically Gaussian model (HGM) as a proxy for any non-Gaussian noise. This model can represent the skewness and/or heavy tail that are typical characteristics of the non-Gaussian noise. Such representation allows us to exploit a CLGM, which is analytically tractable using a KF. This in turn, leads to a RBPF framework for the online inference task. Since within the RBPF framework, PF is confined to a space of lower dimension, it acts as an enabler for scaling to high dimensional problems. The proposed framework here uses a bank of KFs, which is simple to understand; the sophistication comes in the way how we mix and propagate the output of different KFs to get the target filter density. Although this RBPF framework is not entirely new [32], a proper specification of the transition kernel for the state, targeted by PF is often difficult and this technical aspect has received little attention. We elaborate this issue in details and address the proper transition kernel for certain class of noises (where widely used Students' *t* is a special case).

For the second stage, we point how this RBPF framework is already doing a noise adaptive filtering. However to make the filter robust against any large noise deviation, we need prior knowledge about such deviation. Often this is reasonably well known for many practical applications. For such cases, we outline how this information can be encoded in the noise prior. This completes our noise robust online inference framework.

D. *Organization of the article*

The rest of the article is organized as follows. We start with brief descriptions on HGM in section II and simulation based online filtering in section III. This is then followed by the development of the proposed inference framework for LDS under stationary non-Gaussian environment in section IV. We then describe how the above framework can be used for robust noise adaptive filtering in section V. Subsequently, the numerical experiments are shown in VI, which is followed by concluding remarks in section VII.

II. Hierarchical Gaussian model (HGM) for non-Gaussian density

Non-Gaussian densities are often characterized by the presence of heavy tail and/or skewness. Many such densities can often be modeled as hierarchically Gaussian. That is, the density can be represented as Gaussian, conditionally on an auxiliary random variable, known as the 'mixing parameter'. We outline below a brief description for such hierarchical Gaussian representation. For the clarity of the presentation, we consider two separate classes based on the symmetry of underlying density. When the density is symmetric, HGM is represented as scale mixture of normal (SMN), also known as normal variance mixture (NVM). For the non symmetric density, it is represented as normal variance-mean mixture (NVMM) model.

A. Scale mixtures of normal

A random vector X has a scale mixtures of normal density, if it can be expressed as follows

$$X = \mu + \kappa^{\frac{1}{2}}(\Lambda)Z, \tag{5}$$

where μ is a location vector, Z is distributed according to a zero mean multivariate normal with covariance matrix Σ and κ is a positive weight function. Λ is a random variable, known as mixing parameter and is distributed on the positive real line, independent of Z. Note here that conditioned on $\Lambda = \lambda$, X follows a multivariate normal distribution with mean vector μ and covariance matrix $\kappa(\lambda)\Sigma$, i.e., $(X|\Lambda = \lambda) \sim \mathcal{N}(x|\mu, \kappa(\lambda)\Sigma)$. Then, the probability density function (pdf) of X can be expressed as

$$p(x) = \int_0^\infty \mathcal{N}(x|\mu, \kappa(\lambda)\Sigma)p(\lambda)d\lambda, \tag{6}$$

where $p(\lambda)$ is the density function of Λ. $p(\lambda)$ is referred to as the mixing density of SMN representation.

The symmetrical class of SMN includes among others, the popular Students' *t*, the Pearson type VII family, the Slash and the variance gamma distributions[2]. All these distributions are characterized by their heavy tails as compared to the normal distribution. Although there exists many other important SMN distributions (e.g., exponential power, symmetric α -stable, logistic, horse shoe, symmetric generalized hyperbolic distribution) [34], [35], in the sequel, we present only the above mentioned special cases, as the associated mixing densities have computationally attractive form.

[2] The Gaussian distribution in this context, can be thought as a degenerate mixture [33]

1) Multivariate Students' t distribution: When X follows a multivariate Students' t distribution with location μ, scale Σ and degrees of freedom ν (i.e., $X \sim \mathcal{T}(\mu, \Sigma; \nu)$), the pdf of X can be expressed in the following SMN form

$$p(x) = \int_0^\infty \mathcal{N}(x|\mu, \frac{\Sigma}{\lambda}) Ga(\lambda|\frac{\nu}{2}, \frac{\nu}{2}) d\lambda, \tag{7}$$

where $Ga(\cdot|a, b)$ is the gamma density function of the form

$$Ga(\lambda|a, b) = \frac{b^a}{\Gamma(a)} \lambda^{a-1} e^{-b\lambda}, \quad \lambda, a, b > 0, \tag{8}$$

and $\Gamma(a)$ is the gamma function with argument $a > 0$. Consequently, X can be presented in the following hierarchical form

$$X|(\mu, \Sigma, \nu, \lambda) \sim \mathcal{N}(\mu, \lambda^{-1}\Sigma), \quad \lambda|\nu \sim Ga(\frac{\nu}{2}, \frac{\nu}{2}). \tag{9}$$

2) Pearson type VII distribution: If X belongs to the Pearson type VII family, the associated density is given by

$$p(x|\mu, \Sigma, \nu, \delta) = \frac{1}{B(\frac{\delta}{2}, \frac{1}{2})\sqrt{\nu\Sigma}} \big(1 + \frac{(x-\mu)^2}{\nu\Sigma}\big)^{-(\delta+1)/2}, \tag{10}$$

where μ and Σ are the location and scale parameters, $\nu > 0$ and $\delta > 0$ are the shape parameters and $B(a, b)$ is the beta function with arguments $a > 0$ and $b > 0$. The Pearson type VII density can be expressed hierarchically as

$$X|(\mu, \Sigma, \nu, \delta, \lambda) \sim \mathcal{N}(\mu, \lambda^{-1}\Sigma), \quad \lambda|(\nu, \delta) \sim Ga(\frac{\nu}{2}, \frac{\delta}{2}). \tag{11}$$

We get back to Students't distribution when $\nu = \delta$ and Cauchy if $\nu = \delta = 1$.

3) Multivariate slash distribution: The Slash distribution can be hierarchically represented as

$$X|(\mu, \Sigma, \nu) \sim \mathcal{N}(\mu, \lambda^{-1}\Sigma), \quad \lambda|\nu \sim Be(\nu, 1), \tag{12}$$

with $0 < \lambda < 1$ and $\nu > 0$, where $Be(\cdot)$ denotes the Beta distribution.

4) Multivariate variance gamma distribution: When X follows a multivariate variance gamma (VG) distribution with location μ, scale Σ and degrees of freedom $\nu > 0$ (i.e., $X \sim \mathcal{VG}(\mu, \Sigma; \nu)$), the density can be represented in the following hierarchical form

$$X|(\mu, \Sigma, \nu, \lambda) \sim \mathcal{N}(\mu, \lambda^{-1}\Sigma), \quad \lambda|\nu \sim IG(\frac{\nu}{2}, \frac{\nu}{2}), \tag{13}$$

where $IG(a, b)$ is the inverse gamma density given by

$$IG(\lambda|a, b) = \frac{b^a}{\Gamma(a)} \lambda^{-(a+1)} e^{-b/\lambda}. \tag{14}$$

When $\nu = 2$, VG turns into a Laplace distribution.

B. Normal variance-mean mixture

A random vector X is following a normal variance-mean mixture distribution, if it can be expressed as follows [36]

$$X = \mu + \beta\Lambda + \sqrt{\Lambda}Z, \tag{15}$$

where Λ is independent of Z and Z is distributed according to a zero mean multivariate normal with covariance matrix Σ. The random variable Λ is the mixing parameter and is distributed on the positive real line and $\mu, \beta \in \mathbb{R}$. The distribution of X conditioned on $\Lambda = \lambda$ is multivariate normal given as $(X|\Lambda = \lambda) \sim \mathcal{N}(x|\mu + \beta\lambda, \Sigma\lambda)$. Note that when $\beta = 0$, the NVMM turns into a SMN with $\kappa(\lambda) = \lambda$. We describe below some special cases of this class of distributions:

1) Generalized hyperbolic distribution: X in (15) follows a generalized hyperbolic (GH) distribution, if Λ is distributed according to a generalized inverse Gaussian (GIG) distribution as

$$p(\lambda) = \frac{(a/b)^{p/2}}{2K_p(\sqrt{ab})}\lambda^{p-1}exp\{-\frac{a\lambda + b/\lambda}{2}\},\ \lambda > 0, \tag{16}$$

where K_p is the modified Bessel function of the second kind, $a, b \in \mathbb{R}^+$, and $p \in \mathbb{R}$.

2) GH skew Students' t distribution: The hierarchical structure of GH skew Students' *t* distribution is given as [37]

$$(X|\Lambda = \lambda) \sim \mathcal{N}(x|\mu + \beta\lambda, \Sigma\lambda),\ \lambda|\nu \sim IG(\nu/2, \nu/2), \tag{17}$$

where inverse-gamma density is given by (14). Note that the GH skew Students' *t* is a special case of the GH distribution, where the parameters for the GIG are selected as $a = -\nu/2$ $(\nu > 0)$, $b = \sqrt{\nu}$ and $p = 0$. Moreover, when $\beta = 0$, this turns to a symmetric Students' *t* distribution and if $\nu \to \infty$, this becomes a skew normal distribution.

3) GH variance gamma distribution: The hierarchical structure of GH variance gamma distribution is given as

$$(X|\Lambda = \lambda) \sim \mathcal{N}(x|\mu + \beta\lambda, \Sigma\lambda),\ \lambda|\nu \sim Ga(\nu/2, \nu/2), \tag{18}$$

where the gamma density is given by (8).

Remark 1: note that the mixing parameter Λ in II-A and II-B being a random variable, dimension of Λ is always less than or equal to that of random vector X. This observation is important, as we will see later that the PF can be used to target the low dimensional Λ in a RBPF framework.

III. Simulation based online filtering

When a dynamic system (state space model) is nonlinear and/or driven by non-Gaussian noises, the filter density $p(x_k|y_{1:k})$ is in general, analytically intractable. To deal with, many approximated methods have been proposed over time [38]. Particle filtering (PF) is one such method, which uses Monte Carlo simulations to address the filtering problem. In this section, we give a very brief overview of PF and RBPF.

A. Particle filtering (PF)

In PF, the posterior distribution associated with the density $p(x_{0:k}|y_{1:k})$ is approximated by an empirical distribution induced by a set of $N(\gg 1)$ weighted particles (samples) as

$$\widehat{P}_N(dx_{0:k}|y_{1:k}) = \sum_{i=1}^{N} \widetilde{w}_k^{(i)} \delta_{x_{0:k}^{(i)}}(dx_{0:k}); \quad \widetilde{w}_k^{(i)} \geq 0, \tag{19}$$

where $\delta_{x_{0:k}^{(i)}}(A)$ is a Dirac measure for a given $x_{0:k}^{(i)}$ and a measurable set A, and $\widetilde{w}_k^{(i)}$ is the associated weight attached to each particle $x_{0:k}^{(i)}$, such that $\sum_{i=1}^{N} \widetilde{w}_k^{(i)} = 1$. Even though the distribution $\widehat{P}_N(dx_{0:k}|y_{1:k})$ does not admit a well defined density with respect to the Lebesgue measure, we use notational abuse to represent the associated empirical density as

$$\widehat{p}_N(x_{0:k}|y_{1:k}) = \sum_{i=1}^{N} \widetilde{w}_k^{(i)} \delta(x_{0:k} - x_{0:k}^{(i)}). \tag{20}$$

Although (20) is not mathematically rigorous, it is intuitively easier to follow than the stringent measure theoretic notations, especially if we are not concerned with the theoretical convergence studies.

Note that the posterior $p(x_{0:k}|y_{1:k})$, which we target using a PF, is unknown. The empirical distribution $\widehat{P}_N(dx_{0:k}|y_{1:k})$ in (19) is obtained by first generating samples $x_{0:k}^{(i)}$ from a proposal distribution $\pi(x_{0:k}|y_{1:k})$ and then the corresponding weights are obtained using the idea of importance sampling as

$$w_k^{(i)} = \frac{p(x_{0:k}^{(i)}|y_{1:k})}{\pi(x_{0:k}^{(i)}|y_{1:k})}; \; \widetilde{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{j=1}^{N} w_k^{(j)}}. \tag{21}$$

Given this PF output, one can now approximate the marginal distribution $p(x_k|y_{1:k})$ as $\widehat{P}_N(dx_k|y_{1:k}) = \sum_{i=1}^{N} \widetilde{w}_k^{(i)} \delta_{x_k^{(i)}}(dx_k)$. Suppose at time $(k-1)$, we have a weighted particle approximation of the posterior $p(x_{0:k-1}|y_{1:k-1})$ as $\widehat{P}_N(dx_{0:k-1}|y_{1:k-1}) = \sum_{i=1}^{N} \widetilde{w}_{k-1}^{(i)} \delta_{x_{0:k-1}^{(i)}}(dx_{0:k-1})$. Now with a new measurement y_k, we wish to approximate $p(x_{0:k}|y_{1:k})$ with a new set of particles (samples). A standard PF uses the following posterior path-space recursion

$$p(x_{0:k}|y_{1:k}) \propto p(y_k|x_{0:k}, y_{1:k-1}) p(x_k|x_{0:k-1}, y_{1:k-1})\, p(x_{0:k-1}|y_{1:k-1}). \tag{22}$$

Now for the Markovian state space model, this becomes

$$p(x_{0:k}|y_{1:k}) \propto p(y_k|x_k)\, p(x_k|x_{k-1})\, p(x_{0:k-1}|y_{1:k-1}). \tag{23}$$

Assuming that the proposal distribution can be decomposed as $\pi(x_{0:k}|y_{1:k}) = \pi(x_k|x_{0:k-1}, y_{1:k})\pi(x_{0:k-1}|y_{1:k-1})$, we can now implement a sequential importance sampling, where the particles are propagated to time k by sampling a new state $x_k^{(i)}$ from the marginal proposal kernel $\pi(x_k|x_{0:k-1}^{(i)}, y_{1:k})$ and setting $x_{0:k}^{(i)} \triangleq \left(x_{0:k-1}^{(i)}, x_k^{(i)}\right)$. Subsequently using (23), the corresponding weights of the particles can be given by

$$w_k^{(i)} \propto \frac{p(y_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{\pi(x_k^{(i)}|x_{0:k-1}^{(i)}, y_{1:k})}\widetilde{w}_{k-1}^{(i)}; \quad \widetilde{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{j=1}^{N} w_k^{(j)}}. \tag{24}$$

To avoid carrying trajectories with small weights and to concentrate upon the ones with large weights, the particles need to be resampled regularly. The effective sample size N_{eff}, a measure of how many particles that actually contributes to the approximation of the distribution, is often used to decide when to resample. When N_{eff} drops below a specified threshold, resampling is performed. Many efficient resampling schemes have been proposed in the literature. Instead of going into the details, we refer the interested readers to [24], [29], [39]–[41] for a more general introduction to PF.

Remark 2: If one uses the state transition density as proposal with resampling at each step, the corresponding PF is known as *bootstrap particle filter*. This is easy to implement, so very popular in practice.

B. Rao-Blackwellized particle filtering (RBPF)

Although PF is very popular and has been around for a while, it is computationally demanding and notably, it has severe limitations when scaling to higher dimensions [30]. For certain models, when part of the state space is (conditionally) tractable, it is then sufficient to employ a PF for the remaining intractable part of the state space. If it is possible to exploit such analytical substructure, the Monte Carlo based estimation is then confined to a space of lower dimension. As a consequence, the estimate obtained is often better (in terms of asymptotic variance) and never worse than the estimate provided by the PF targeting the full state space. The resulting method is popularly known as *Rao-Blackwellized particle filtering* [32], [42]–[45]. Note that solving part of the state vector analytically (or using analytical approximations) leaves the remaining part to be targeted only by PF. Thus RBPF acts as a practical enabler for scaling to high dimensional problems.

IV. Inference for LDS under stationary non-Gaussian environment

Consider the LDS in (1), driven by known stationary non-Gaussian noises, assumed to be hierarchically Gaussian as in Section II. Given the model and assuming that the initial prior, $p(x_0)$ is a known Gaussian density[3], our objective is to recursively target the intractable filtering density $p(x_k|y_{1:k})$.

Let at time step k, λ_k^w and λ_k^e be the corresponding mixing parameters for w_k and e_k. We define an auxiliary vector $\lambda_k \equiv (\lambda_k^w,\ \lambda_k^e)^T$. Then the noises w_k and e_k are hierarchically Gaussian as

$$w_k|\lambda_k \sim \mathcal{N}\big(\mu_k^w(\lambda_k), Q_k(\lambda_k)\big), \tag{25a}$$

$$e_k|\lambda_k \sim \mathcal{N}\big(\mu_k^e(\lambda_k), R_k(\lambda_k)\big), \tag{25b}$$

where $\mu_k^w,\ \mu_k^e$ are the corresponding mean and Q_k and R_k are the corresponding covariance of the hierarchical Gaussian noises[4]. These parameters can possibly depend upon λ_k. Such a noise representation admits a CLGM, which is analytically tractable using the KF. This opens up a RBPF implementation for the online inference problem. In the sequel, we describe this RBPF framework. This section is organized as follows. We first outline one complete iteration of the proposed RBPF framework, which is followed by the specification of the dynamics for the state (i.e., mixing parameters), targeted by associated PF. Next, an algorithmic description of the approach is provided. This is then followed by the descriptions on the likelihood function estimation and p-step ahead prediction.

A. *Posterior propagation cycle of RBPF*

At time zero, $p(x_0)$ is known. Suppose at time step $(k-1)$, we have the joint target distribution $p(x_{k-1}, \lambda_{1:k-1}|y_{1:k-1})$. This can be decomposed as

$$p\left(\begin{matrix} x_{k-1}, \\ \lambda_{1:k-1} \end{matrix} \,\middle|\, y_{1:k-1}\right) = p\left(x_{k-1} \,\middle|\, \begin{matrix} \lambda_{1:k-1}, \\ y_{1:k-1} \end{matrix}\right) p\Big(\lambda_{1:k-1}|y_{1:k-1}\Big), \tag{26}$$

where $p(\lambda_{1:k-1}|y_{1:k-1})$ is targeted by a PF and is empirically given by a set of $N(>> 1)$ weighted random particles as

$$p(\lambda_{1:k-1}|y_{1:k-1}) = \sum_{i=1}^{N} \widetilde{w}_{k-1}^{(i)} \delta(\lambda_{1:k-1} - \lambda_{1:k-1}^{(i)}), \tag{27}$$

[3] More generally, $p(x_0)$ can be taken to be a hierarchically Gaussian density.

[4] Instead of both the noises being non-Gaussian, if one of them is Gaussian, the inference framework, described subsequently, is still valid. In that case λ_k becomes an auxiliary random variable, representing the mixing parameter of the non-Gaussian noise.

with $\widetilde{w}_{k-1}^{(i)} \geq 0$ and $\sum_{i=1}^{N} \widetilde{w}_{k-1}^{(i)} = 1$. Using (27), we have

$$p(\lambda_{k-1}|y_{1:k-1}) = \sum_{i=1}^{N} \widetilde{w}_{k-1}^{(i)} \delta(\lambda_{k-1} - \lambda_{k-1}^{(i)}). \tag{28}$$

Noting that, given a sequence $\lambda_{1:k-1}^{(i)}$, the dynamic system now becomes a CLGM. So $p(x_{k-1}|\lambda_{1:k-1}^{(i)}, y_{1:k-1})$ can be obtained by a KF, given by

$$p(x_{k-1}|\lambda_{1:k-1}^{(i)}, y_{1:k-1}) = \mathcal{N}\big(\widehat{x}_{k-1}(i), P_{k-1}(i)\big), \tag{29}$$

where for notational clarity, we suppress the dependency of the parameters on $\lambda_{1:k-1}^{(i)}$. Since a KF is running for each sequence (henceforth denoted by an index i), we have total N number of KFs running in parallel at any given time. Now using (27) and (29) together, the filter distribution $p(x_{k-1}|y_{1:k-1})$ can be obtained as

$$\begin{aligned} p(x_{k-1}|y_{1:k-1}) &= \int p\left(x_{k-1} \,\middle|\, \begin{matrix} \lambda_{1:k-1}, \\ y_{1:k-1} \end{matrix} \right) \\ &\quad \times p\left(\lambda_{1:k-1} \,\middle|\, y_{1:k-1} \right) d\lambda_{1:k-1}, \\ &\approx \sum_{i=1}^{N} \widetilde{w}_{k-1}^{(i)} \; \mathcal{N}\left(\widehat{x}_{k-1}(i), P_{k-1}(i) \right), \end{aligned} \tag{30}$$

which is a weighted (finite) mixture of N Gaussian distributions. The mean and covariance of this filter distribution (assuming them to be finite) can be given as

$$\widehat{x}_{k-1} = \sum_{i=1}^{N} \widetilde{w}_{k-1}^{(i)} \widehat{x}_{k-1}(i), \tag{31a}$$

$$P_{k-1} = \sum_{i=1}^{N} \widetilde{w}_{k-1}^{(i)} \big\{ P_{k-1}(i) + \big(\widehat{x}_{k-1} - \widehat{x}_{k-1}(i)\big)\big(\,\cdot\,\big)^{T} \big\}, \tag{31b}$$

where $(A)(\cdot)^T$ is a shorthand for AA^T. Now having observed y_k, we would like to propagate the joint posterior in (26) to time k, i.e.,

$$p(x_{k-1}, \lambda_{1:k-1}|y_{1:k-1}) \xrightarrow{y_k} p(x_k, \lambda_{1:k}|y_{1:k}). \tag{32}$$

This can be achieved in the following steps ((1)-(4)) as described below:

*1) **PF prediction step**:* Generate N new samples $\lambda_k^{(i)}$ from appropriately chosen proposal $\pi(\lambda_k|\lambda_{1:k-1}^{(i)}, y_{1:k})$. Then set $\lambda_{1:k}^{(i)} = \{\lambda_{1:k-1}^{(i)}, \lambda_k^{(i)}\}$, for $i = 1, \cdots, N$, representing the particle trajectories up to time k. For proposal selection, see Remark 4 below.

*2) **KF prediction step**:* Since the prior $p(x_{k-1}|\lambda^{(i)}_{1:k-1}, y_{1:k-1})$ for this step is Gaussian (given by (29)) and the noises are also Gaussian given $\lambda^{(i)}_k$, (see (25)), the dynamic system is now linear-Gaussian conditioned on $\lambda^{(i)}_{1:k}$. So, using KF, the predictive distribution can be obtained analytically, which is also Gaussian and given as $p(x_k|\lambda^{(i)}_{1:k}, y_{1:k-1}) = \mathcal{N}\big(\widehat{x}_{k|k-1}(i), P_{k|k-1}(i)\big)$, where

$$\widehat{x}_{k|k-1}(i) \;=\; A_k\ \widehat{x}_{k-1}(i) + B_k\ \mu^w_k(i) \tag{33}$$

$$P_{k|k-1}(i) \;=\; A_k\ P_{k-1}(i)\ A_k^T + B_k\ Q_k(i)\ B_k^T. \tag{34}$$

*3) **KF update step**:* Suppose we have now the new observation y_k. We can now update to the posterior distribution $p(x_k|\lambda^{(i)}_{1:k}, y_{1:k})$, which is also Gaussian due to the CLGM, denoted as

$$p(x_k|\lambda^{(i)}_{1:k}, y_{1:k}) = \mathcal{N}\big(\widehat{x}_k(i), P_k(i)\big). \tag{35}$$

The parameters $\widehat{x}_k(i)$, $P_k(i)$ can be obtained from the KF recursively, using the following steps:

$$\begin{aligned} \widehat{x}_k(i) \;&=\; \widehat{x}_{k|k-1}(i) + \\ &\qquad K_k(i)\big\{y_k - C_k\ \widehat{x}_{k|k-1}(i) - \mu^e_k(i)\big\} \end{aligned} \tag{36}$$

$$P_k(i) \;=\; P_{k|k-1}(i) - K_k(i)\ C_k^T\ P_{k|k-1}(i), \tag{37}$$

where

$$K_k(i) \;=\; P_{k|k-1}(i)\ C_k^T\ S_k^{-1}(i) \tag{38}$$

$$S_k(i) \;=\; C_k\ P_{k|k-1}(i)\ C_k^T + R_k(i). \tag{39}$$

Moreover, the marginal likelihood is also obtained in closed form, which is also Gaussian and is given by

$$p(y_k|\lambda^{(i)}_{1:k}, y_{1:k-1}) = \mathcal{N}\big(\mu^L_k(i),\ \Sigma^L_k(i)\big), \tag{40}$$

where

$$\mu^L_k(i) \;=\; C_k\ \widehat{x}_{k|k-1}(i) + \mu^e_k(i) \tag{41a}$$

$$\Sigma^L_k(i) \;=\; S_k(i). \tag{41b}$$

*4) **PF update step**:* Now given the observation y_k and the particles $\{\lambda^{(i)}_{1:k}\}_{i=1}^N$, we need to update to the posterior

$$p(\lambda_{1:k}|y_{1:k}) = \sum_{i=1}^N \widetilde{w}^{(i)}_k \delta(\lambda_{1:k} - \lambda^{(i)}_{1:k});\ \ \widetilde{w}^{(i)}_k \geq 0, \tag{42}$$

with $\sum_{i=1}^{N} \widetilde{w}_k^{(i)} = 1$. The corresponding weight $\widetilde{w}_k^{(i)}$ can be obtained (using (22)) as

$$w_k^{(i)} \propto \frac{p(y_k|\lambda_{1:k}^{(i)}, y_{1:k-1}) p(\lambda_k^{(i)}|\lambda_{1:k-1}^{(i)}, y_{1:k-1})}{\pi(\lambda_k^{(i)}|\cdot)} \widetilde{w}_{k-1}^{(i)}, \tag{43a}$$

$$\widetilde{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{j=1}^{N} w_k^{(j)}}. \tag{43b}$$

The density $p(y_k|\lambda_{1:k}^{(i)}, y_{1:k-1})$ is already obtained in (40). For now we assume that $p(\lambda_k^{(i)}|\lambda_{1:k-1}^{(i)}, y_{1:k-1})$ is specified, which we discuss further in IV-B.

This completes one propagation cycle. To propagate to the next cycle, we first resample the particles $\lambda_{1:k}^{(i)}$ (and also the associated $\left\{\widehat{x}_k(i),\ P_k(i)\right\}$) whenever necessary, before following again the steps (1)-(4).

Remark 3: so long we can directly generate samples from w_k and e_k, we can in principle, target the same inference task e.g., using a PF. However, when the state space is high dimensional, PF is well known to be inefficient. On the other hand, in our RBPF framework, PF targets only low dimensional state (auxiliary mixing vector, λ_k); conditionally the remaining state vector are treated using KF. As a result, RBPF can scale well with the dimensions here.

B. Specification of dynamics for λ_k

As PF is essentially designed for the dynamically evolving model, we need to specify a proper transition density $p(\lambda_k|\lambda_{1:k-1}, y_{1:k-1})$, describing the time evolution of the path space $\lambda_{1:k}$. However this transition density is typically unknown and the only information we have is that λ_k is distributed according to a (known) stationary density (say, p^*). We note that earlier in the context of symmetric α-stable noise, [46], [47] treated λ_k as unknown parameter with $p(\lambda_k|\lambda_{1:k-1}, y_{1:k-1}) = p(\lambda_k) \equiv p^*$ (i.e., λ_k are generated independently over k). In the PF context, this is completely arbitrary and fundamentally weak, as path space based recursion (see e.g., (43a)) requires a transition kernel, which links any newly generated particle to its ancestral lineage. To our knowledge, a provable state transition density in this context is not addressed adequately in the literature.

In our approach, we specify the transition density through a Markov kernel such that the samples generated from this kernel constitute a Markov chain with the invariant distribution as marginal; since this invariant distribution is already known, we initialize the chain with this invariant distribution. In doing so, we follow [48], where the authors construct an AR(1) process using the *auxiliary latent variables*, such that the resulting time series is stationary with known marginals. Moreover, as shown in the original article, the time series model follows simple auto-correlation function, which decays exponentially. This property is in fact very beneficial as it can limit the dependency of the sufficient statistics of $p(x_k|\lambda_{1:k}, y_{1:k})$ on the particle path space.

As shown in [48], based on the known invariant marginal, one can construct such a transition kernel easily, when $p(\lambda_k)$ belongs to infinitely divisible convolution-closed exponential family and a family of density functions that includes among others, (inverse) gamma, normal and beta densities. To keep the description simple, rather than going into the further details, we describe below one example to illustrate the idea. For more details, the reader may refer to [48].

1) Example: note from II-A4, if X is distributed according to a multivariate variance gamma, then the marginal $p(\lambda)$ is following an inverse gamma given by (13)-(14). Suppose we are interested in constructing $p(\lambda_k|\lambda_{k-1})$. This can be implicitly obtained using the following relation [48]

$$\lambda_k = \frac{\nu/2 + U_k \ \lambda_{k-1}}{V_k}, \tag{44}$$

where $U_k \sim Ga(\alpha, 1)$ and $V_k \sim Ga(\nu/2 + \alpha, 1)$, which are independent and also independent of each other. We also have

$$E(\lambda_k|\lambda_{k-1}) = \mu(1-\rho) + \rho\lambda_{k-1}, \tag{45}$$

where $\rho = \frac{2\alpha}{\nu+2\alpha-2}$ is the autocorrelation function[5] and $\mu = \frac{\nu}{\nu-2}$ is the mean of the marginal distribution. Note that $E(\lambda_k|\lambda_{k-1})$ exists for $\nu > 2$ and the autocorrelation function of λ_k is $\rho(\tau) = \rho^\tau$. Thus, the transition kernel can be specified by a first order AR model with inverse-gamma marginals. The unknown parameter α (associated with U_k and V_k) is specified here through the selected value of ρ.

Remark 4: The optimal proposal [49] given by $\pi(\lambda_k|\lambda_{1:k-1}, y_{1:k}) = \pi(\lambda_k|\lambda_{k-1}, y_k)$ is often unavailable in practice. One simple alternative is to implement a *bootstrap* PF with transition kernel $\pi(\lambda_k|\lambda_{k-1})$ as the proposal.

C. Algorithmic summary

A *bootstrap* implementation of the proposed RBPF for the LDS with hierarchical Gaussian noises are summarized in Algorithm 1.

D. Estimation of the likelihood function

The likelihood function is the joint density of the observation $y_1, \cdots, y_T$, which can be decomposed as

$$p(y_1, \cdots, y_T) = \prod_{k=1}^{T} p(y_k|y_{1:k-1}), \tag{46}$$

[5] The autocorrelation function is restricted to be positive (i.e., $\rho \in (0,1)$) for such construction.

Algorithm 1 RBPF for LDS

```
Set HGM for process and/or measurement noises
- $p(\lambda_k)$ is set accordingly
For each particle $i = 1, \dots, N$ do
Initialization:
  • set $p(x_0^{(i)})$ for each KF
Iterations:
For $k = 1, 2, \dots$ do
  1) PF prediction step
     • sample $\lambda_k^{(i)}$ according to section IV-B
        – if $k = 1$, $\lambda_k^{(i)} \sim p(\lambda_k)$
        – else, $\lambda_k^{(i)} \sim p(\lambda_k | \lambda_{k-1}^{(i)})$
     • set $\lambda_{1:k}^{(i)} \triangleq (\lambda_{1:k-1}^{(i)}, \lambda_k^{(i)})$
  2) KF prediction step
     • set $\mu_k^w(i), \mu_k^e(i), Q_k(i), R_k(i)$ (Eq. (25))
     • compute $\widehat{x}_{k|k-1}(i)$ and $P_{k|k-1}(i)$ (Eq. (33)-(34))
  3) KF update step
     • compute $\widehat{x}_k(i)$ and $P_k(i)$ (Eq. (36)-(39))
     • compute $p(y_k | \lambda_{1:k}^{(i)}, y_{1:k-1})$ (Eq. (40)-(41))
  4) PF update step
     • weight update
        – $w_k^{(i)} \propto p(y_k | \lambda_{1:k}^{(i)}, y_{1:k-1})$
        – normalize weights
     • resample the particles
        – let resampled particle indices be $j_i \in \{1, \cdots, N\}$
        – set $\widehat{x}_k(i) = \widehat{x}_k(j_i)$ and $P_k(i) = P_k(j_i)$
        – set $\lambda_{1:k}^{(i)} = \lambda_{1:k}^{(j_i)}$
        – move to the next step
```

where for $k=1$, $p(y_k|y_{1:k-1})$ is interpreted as $p(y_1|\varnothing) \equiv p(y_1)$. The density $p(y_k|y_{1:k-1})$ can be obtained by marginalizing $\lambda_{1:k}$ in (40) using (42) as

$$p(y_k|y_{1:k-1}) \approx \sum_{i=1}^{N} \widetilde{w}_k^{(i)} \mathcal{N}\big(\mu_k^L(i),\ \Sigma_k^L(i)\big). \tag{47}$$

E. p-step ahead prediction

Suppose at time step k, we have the observations $\{y_1, \cdots, y_k\}$ and the corresponding filter density $p(x_k|y_{1:k})$. The p-step ahead prediction of the state can now be obtained as follows:

$$\begin{aligned} p(x_{k+p}|y_{1:k}) &= \int p(x_{k+p}|y_{1:k}, \lambda_{1:k})\ p(\lambda_{1:k}|y_{1:k}) d\lambda_{1:k} \\ &\approx \sum_{i=1}^{N} w_k^{(i)} p(x_{k+p}|y_{1:k}, \lambda_{1:k}^{(i)}). \end{aligned} \tag{48}$$

Now $p(x_{k+p}|y_{1:k}, \lambda_{1:k}^{(i)})$ can be obtained using p-step ahead KF prediction as

$$p(x_{k+p}|y_{1:k}, \lambda_{1:k}^{(i)}) = \mathcal{N}(\mu_{k+p|k}^{(i)}, \Sigma_{k+p|k}^{(i)}), \tag{49}$$

with

$$\mu_{k+p|k}^{(i)} = \Phi_{k+1+p,k+1}\ \widehat{x}_k(i)\ +\ \sum_{j=1}^{p} \Phi_{k+1+p,k+1+j}\ B_{k+j}\ \mu_{k+j}^{w}(i), \tag{50}$$

$$\begin{aligned} \Sigma_{k+p|k}^{(i)} = \Phi_{k+1+p,k+1}\ P_k(i)\ (\Phi_{k+1+p,k+1})^T\ + \\ \sum_{j=1}^{p} \Phi_{k+1+p,k+1+j}\ B_{k+j} Q_{k+j}(i) (B_{k+j})^T (\Phi_{k+1+p,k+1+j})^T, \end{aligned} \tag{51}$$

where we use the notation $\Phi_{k,l} = A_{k-1}A_{k-2}\cdots A_l \ \ (k > l)$ and $\Phi_{k,k} = I$ [3].

V. ROBUST NOISE ADAPTIVE FILTERING FOR LDS

Although, the distribution of the noise is assumed to be stationary and known in the previous section, it is important to observe that the proposed algorithm is already doing a noise adaptive filtering. To see this, first note from (28) that we are also estimating λ_k sequentially over time. Now moving from time $(k-1)$ to k, when we generate N new samples $\{\lambda_k^{(i)}\}_{i=1}^N$, the empirical distribution associated with $p(\lambda_{1:k})$ is given by the particle cloud $\big\{\lambda_{1:k}^{(i)},\ (1/N)\big\}_{i=1}^N$. Since the noise in (25) at time k is assumed to be hierarchically Gaussian, we can now represent the density for the noise (say v_k) as

$$p(v_k) \approx \frac{1}{N} \sum_{i=1}^{N} \mathcal{N}\Big(v_k|\mu(\lambda_{1:k}^{(i)}), \Sigma(\lambda_{1:k}^{(i)})\Big), \tag{52}$$

which is an equally weighted finite (random) Gaussian mixture (with component weight $= 1/N$). This serves as our prior for the (unknown) noise. When the new measurement y_k arrives, we update to noise posterior using the Bayes rule. The posterior now consists of the same random mixture components, but the component weights are adjusted according to the observation likelihood (i.e. $p(y_k|\lambda_{1:k}^{(i)}, y_{1:k-1})$). This construction ensures that we have a non-Gaussian noise adaptive filter. However, the stability of this filter is still sensitive to any potential large noise that cannot be accounted by the selected prior. In practice, the robustification for this noise adaptive filter is done through a flat prior selection. This ensures that the resulting filter is robust to outliers, yet at the same time due to its noise adaptive behavior, the performance does not degrade when the outliers are absent.

VI. Numerical studies

We start here with the inference problem when the measurement noise is skewed but its distribution is known. This is illustrated for a stochastic volatility problem (finance). Next we consider the cases where noise parameters are unknown and time varying. Here we test the proposed robust noise adaptive filter on a time series problem for the following cases: (a) unknown measurement noise and (b) both process and measurement noises are unknown. Finally we consider a maneuvering target tracking example and compare the proposed approach with the interacting multiple model (IMM) algorithm.

A. LDS with known stationary non-Gaussian noise

We consider the online filtering problem for the volatility example defined by (3a)-(3b). This example illustrates the strength of the proposed framework in terms of treating a skewed noise (although we note that PF is more appropriate for this scalar problem when computational cost is considered). As is evident from Figure (1), the observation noise is highly skewed. In the literature, this has been approximated e.g., by a mixture of seven Gaussian components by equating the first four moments [50]. Here we approximate this noise density using a GH skew Student's *t* distribution with $\mu = 1.75$, $\beta = -2.3$, $\Sigma = 1$ and $\nu = 5.8$. Following [21], we select $\gamma_0 = 0$, $\gamma_1 = 0.9$ and $\sigma_n^2 = 0.1$ and using (2), generate a (simulated) return data y_k for 1000 time steps. Next, using the simulated data, we estimate the filter distribution by Algorithm1. We observed that the algorithm is handling well the sequential estimation task. The simulated data, the (approximated) observation noise density and a typical realization for the filter mean estimates are plotted in Figure (2).

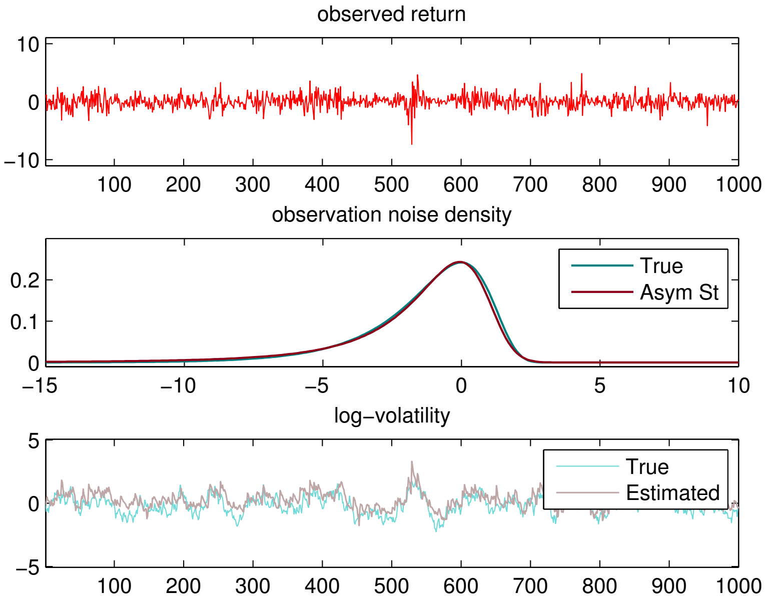


Figure 2: Stochastic volatility example: (top) simulated return data y_k, (middle) observation noise density and its GH skew Student's t approximation, (bottom) the true log-volatility and one realization of the filter mean estimates.

B. *Time varying and unknown noise parameters*

Here we start with the case of unknown measurement noise, which is then followed by the case, where both the process noises and measurement noises are unknown.

1) Measurement noise is unknown: we consider the following time series problem given by

$$s_k = 1.51 s_{k-1} - 0.55 s_{k-2} + w_k, \tag{53a}$$

$$y_k = s_k + e_k, \tag{53b}$$

where the signal s_k is evolving as a second order auto-regressive process, with $w_k \sim \mathcal{N}(0,1)$. The distribution for the measurement noise e_k is assumed to be unknown. The simulated (synthetic) measurements y_k are generated using the mixture distribution $e_k \sim 0.95\mathcal{N}(0,10) + 0.05\mathcal{N}(0,100)$. For the robust noise adaptive filter, the prior for unknown e_k is taken as a zero mean Student's t with $\Sigma = 10^4$ and $\nu = 5$. A typical realization of the filter mean estimate is shown in Figure (3). One can observe that the measurements contain outliers that are appearing sporadically (statistically independent from one another), but the filter is doing a really good job in estimating the hidden signals.

But there may be situation where the outliers can be persistence (time correlated). For example, noise

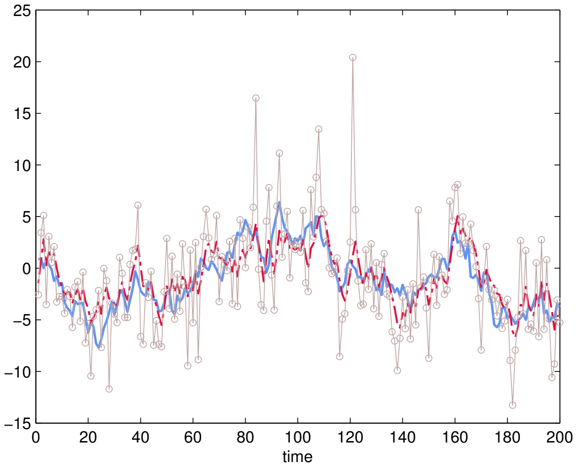


Figure 3: When measurement noises containing sporadic outliers; true signal ('$-$'), measurements ('$-o$') and mean estimates of the signal ('$-.$').

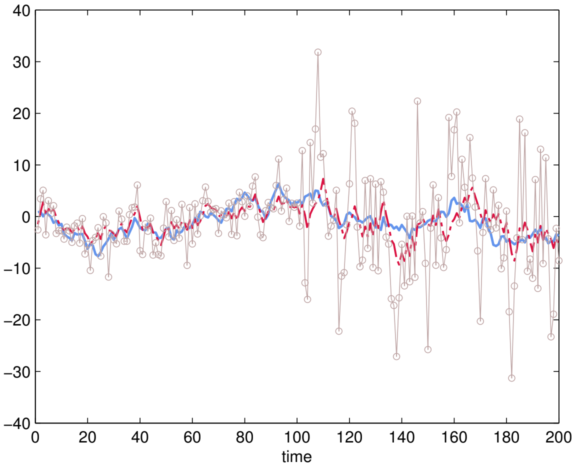


Figure 4: When measurement noises containing persistent outliers; true signal ('$-$'), measurements ('$-o$') and mean estimates of the signal ('$-.$').

may temporarily switch to a different regime (with a higher covariance) due to sensor malfunction. This change in noise cannot be counted normally by a nominally specified noise model. We consider this scenario next. For the simulated observations, we assume that the measurement noises for the first 100 time steps is distributed as $e_k \sim \mathcal{N}(0, 10)$. Then due to the sensor malfunctioning, the measurement noises for the next 100 time steps is distributed as $e_k \sim \mathcal{N}(0, 100)$. However, this knowledge is not available to the filter. Again the prior for unknown e_k is taken to be a zero mean Student's t with $\Sigma = 10^4$ and $\nu = 5$ and a typical realization of the filter mean estimate is shown in Figure (4). In this case as well, the proposed filter is doing a good job in estimating the signals.

2) Both process and measurement noises are unknown: for this problem, although the framework is relatively simple (i.e., use HGM priors for both the noises as in (25)), we note that the inference problem can be difficult, at least for certain problems, due to identifiability issue. For example, if for any running KF (indexed by i), there is any larger than the usual value of innovation (defined as $\left\{y_k - C_k\, \widehat{x}_{k|k-1}(i) - \mu^e_k(i)\right\}$ in (36)), KF cannot immediately distinguish whether this is due to any observation outlier or structural break in the process model. Nonetheless, filtering with unknown heavy tailed process and measurement noises have earlier been considered in e.g., [9], [16], [27] and so we include some numerical studies as well. We consider the LDS as in (53), where the true (but unknown) noises are distributed as $w_k \sim 0.8\mathcal{N}(0,10)+0.2\mathcal{N}(0,100)$ and $e_k \sim 0.9\mathcal{N}(0,100)+0.1\mathcal{N}(0,1000)$. For filtering, the state is estimated using 50 particles and the noise priors as $p(w_k) = \mathcal{T}(0,10^4,5)$ and $p(e_k) = \mathcal{T}(0,10^5,5)$ respectively. The true (generated) noise realizations are shown in Figure (5a) while the true and a typical estimated (mean) state along with measurements are shown in Figure (5b). Next, we consider the case of persistent noise outliers with the same filter settings. Here the process noises for first 40 step is generated with $\mathcal{N}(0,100)$ and the rest with $\mathcal{N}(0,10)$. Similarly the measurement noises for first 20 step is generated with $\mathcal{N}(0,1000)$ and the rest with $\mathcal{N}(0,100)$. The true (generated) noise realizations for this case are shown in Figure (5c). Corresponding true and a typical estimated (mean) state along with measurements are shown in Figure (5d). For both cases, we observe that the filter is performing reasonably well.

C. Tracking a maneuvering target

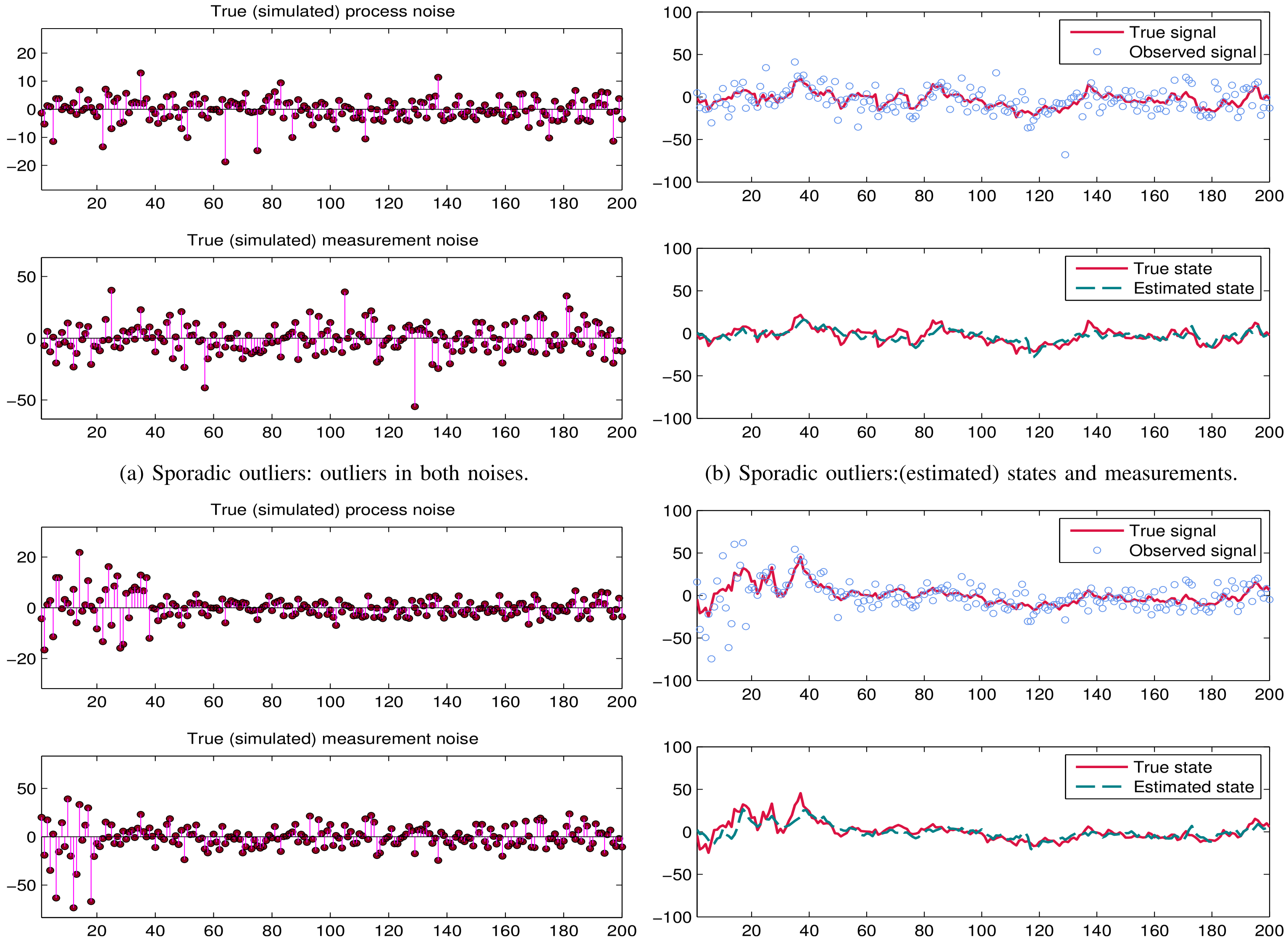
We consider a (simplified) problem of tracking a maneuvering target in two dimensional plane, where the state vector (x_k) consists of the Cartesian position and velocity. The noise corrupted snapshots of the positions (y_k) are available as the measurements. The target starts at $(0,0)$ and initially proceeds with a linear motion. This is followed by a (clockwise) coordinated turn and then again a linear motion. The true target trajectory and the noisy measurements are shown in Figure (6). We propose to track the target first, using a so called constant velocity (CV) model [51] given by

$$x_{k+1} = \begin{bmatrix} I_2 & TI_2 \\ 0 & I_2 \end{bmatrix} x_k + \begin{bmatrix} \frac{T^2}{2}I_2 \\ TI_2 \end{bmatrix} v_k \tag{54}$$

$$y_k = [I_2 \;\; 0]\, x_k + e_k, \tag{55}$$

where T is the sampling time in second (we use $T = 1$), I_2 is a two dimensional identity matrix, v_k and e_k are the process and the measurement noise respectively. We assume that e_k is generated according to a known distribution, given by $e_k \sim \mathcal{N}(0, diag[80^2,\;\, 80^2])$, which is also used to generate the true measurements here. However due to multi-modality in the track behavior, an appropriate process noise is difficult to specify. This point is illustrated by estimating the track using two different process noise standard deviations, i.e., $\sigma_v = 1$ and $\sigma_v = 50$ (assuming v_k to be Gaussian). The results are shown Figure (7a) and Figure (7b) respectively. We see that $\sigma_v = 1$ is a better choice for the (initial) linear part of the trajectory, whereas $\sigma_v = 50$ is more suitable during the maneuvering part. Thus a fixed form of process noise is not appropriate for this problem. Moreover, specification of suitable noise parameters (i.e., σ_v here) is often not obvious.

Subsequently, we try our robust noise adaptive (RBPF) filter. Here the prior for the process noise is taken as a symmetric Laplace $\big(p(v_k) \sim \mathcal{L}(0,10^6)\big)$ and we use 50 particles and $\rho = 0.05$. A typical estimated



(a) Sporadic outliers: outliers in both noises.

(b) Sporadic outliers:(estimated) states and measurements.

(c) Persistent outliers: outliers in both noises.

(d) Persistent outliers:(estimated) states and measurements.

Figure 5: Filtering when both the process and the measurement noises contain outliers

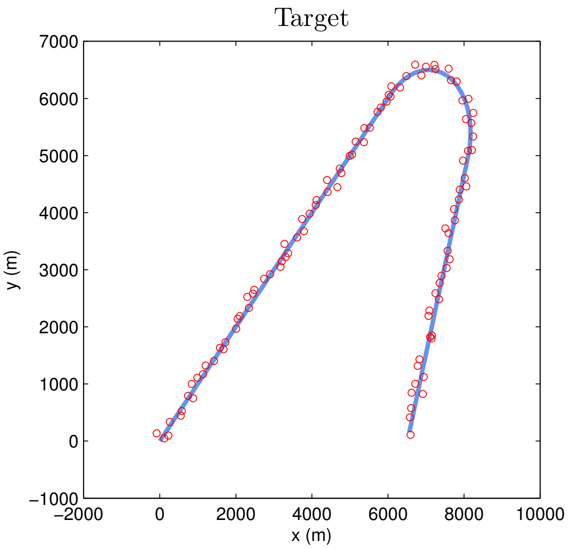
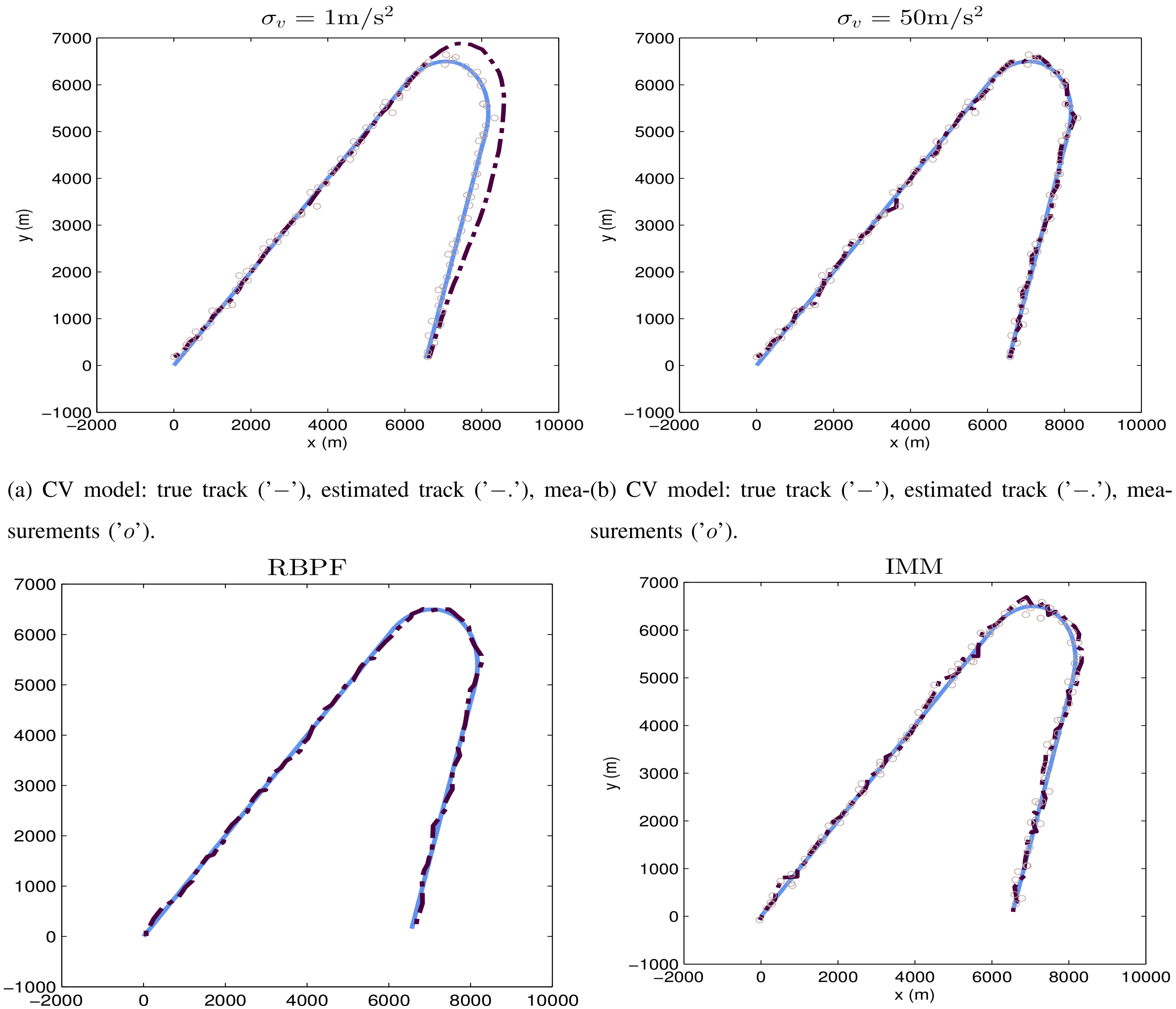


Figure 6: The true positions ('$-$') and the measured positions ('o') of the target (the target starts at (0, 0)).

track is shown in Figure (7c). We also implement an IMM filter [52] with (Gaussian process noise) $\sigma_v = 1$ and $\sigma_v = 50$. We assume that the initial probability for the modes are equal and a mode transition probability matrix $\pi = [0.9\ 0.1;\ 0.1\ 0.9]$. A typical estimate of the track is shown in Figure (7d). We observed that both our RBPF and IMM are performing well for this problem. We also compare them over 20 Monte Carlo runs. The statistics for the maximum absolute errors (max abs err) and the average root mean square error (avg. RMSE) are shown in Table I. From Table I, we see that IMM is performing slightly better, although at the cost of requiring a properly specified mode transition matrix π. In practice, often π is unknown and estimating π online is an arduous task. This problem does not arise in our approach. Finally, there may be outliers in the measurement noise as well e.g., due to occasional sensor malfunctions. This problem cannot be solved trivially by tuning the parameters of the noises in a KF setup. It is hard for any filter to distinguish immediately whether any outlier is arising from process or measurement noise. To test our approach, we keep the same trajectory, but now the measurements are generated from a

	IMM	RBPF
max abs error	164.95	191.87
avg. RMSE	65.33	69.71

Table I: Error statistics over 20 Monte Carlo runs



(a) CV model: true track ('−'), estimated track ('−.'), measurements ('o').

(b) CV model: true track ('−'), estimated track ('−.'), measurements ('o').

(c) Robust noise adaptive model: true track ('−'), estimated track ('−.').

(d) IMM: true track ('−'), estimated track ('−.'), measurements ('o').

Figure 7: Maneuvering target tracking example; true and estimated trajectory of the target

mixture distribution given by $e_k \sim 0.8\mathcal{N}(0, diag[80^2,\ 80^2]) + 0.2\mathcal{N}(0, diag[300^2,\ 300^2])$. We implement our algorithm with priors for the process and the measurement noises to be Laplace and Students' t noise respectively, given by $p(v_k) \sim \mathcal{L}(0, 10^8)$ and $p(e_k) \sim \mathcal{T}(0, 10^8; 4)$. Again we use 50 particles and $\rho = 0.05$. The filter is performing reasonably well in most cases. A typical estimate of the track is shown in Figure (8).

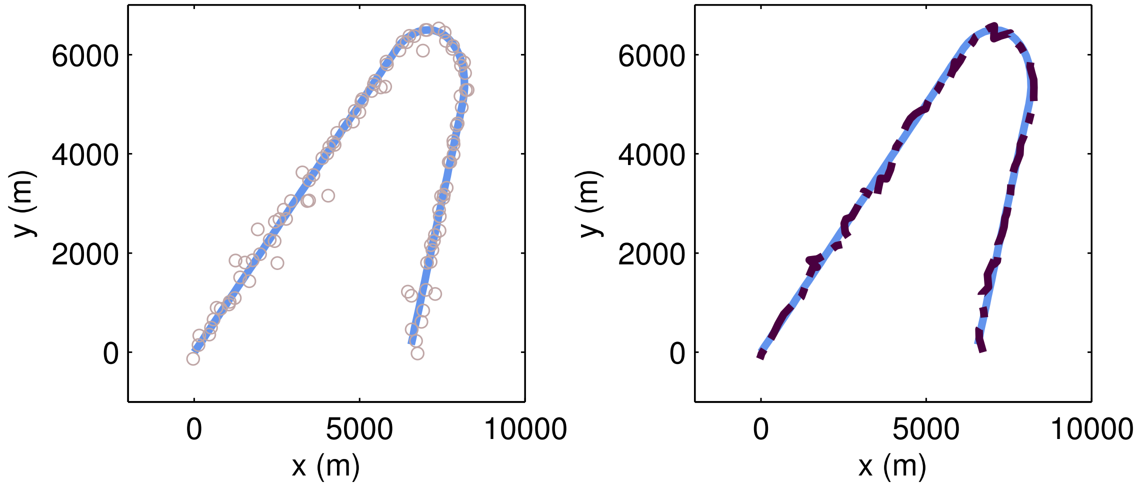


Figure 8: (left) true trajectory with measurements corrupted by outliers, (right) estimated trajectory by robust noise adaptive model

VII. Concluding remarks

This article considers the difficult online inference problems for linear dynamic systems under a very realistic set-up, where (a) the noises can have non-Gaussian densities (either appearing naturally or are modeled to accommodate outliers) and (b) the noise parameters can be unknown and time varying. The corresponding non-Gaussian density is characterized in terms of the skewness and/or the heavy tails, that is commonly observed in many practical applications of interest. We note that unlike the heavy tails, inference with the skewed noise has not attracted considerable research attentions.

For the inference task, we envisage here a new Rao-Blackwellized particle filter by leveraging on a so called hierarchical Gaussian model on the noises. We showed that the proposed framework is not only robust to any noise outlier, but also adaptive to potentially unknown and time varying noise parameters. However, the framework requires a valid transition kernel for the intractable state, targeted by the particle filter. We outlined how such kernels can be constructed, at least for certain classes of noises that cover many commonly occurring (non-Gaussian) noises, where the well explored Students' t is just a special case. The framework essentially runs a bank of (interacting) Kalman filters and so is relatively easy to understand and/or implement. We also explained how the problem can easily scale up with the dimensions using Rao-Blackwellization idea. The proposed algorithm here is very simple and flexible, although a bit computationally demanding. We subsequently carried out numerical experiments including a comparison to IMM. We observed empirically that the framework is doing a very good job even with only 50 particles. The associated tasks of (offline) state smoothing, model parameters learning and also extending the framework for space-time inference are left as future works.

Acknowledgment

The author would like to thank COOP-LOC, funded by SSF and CADICS, funded by Swedish Research Council (VR) for the financial supports.

References

[1] M. West and P. J. Harrison, *Bayesian Forecasting and Dynamic Models*. Springer-Verlag, New York, USA, 1997.

[2] A. Bagchi, *Optimal Control of Stochastic Systems*. Prentice Hall Inc., NJ, USA, 1993.

[3] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, N.J.: Prentice Hall, 1979.

[4] Y. Chen, E. E. Kuruoglu, and H. C. So, "Estimation under additive Cauchy-Gaussian noise using Markov chain Monte Carlo," in *Proceedings of IEEE Workshop on Statistical Signal Processing (SSP)*, 2014.

[5] J. Vila-Valls, Q. Wei, and C. Fernández-Prades, "Robust Gaussian sum filtering with unknown noise statistics: Application to target tracking," in *Proceedings of IEEE Workshop on Statistical Signal Processing (SSP)*, 2014.

[6] A. M. Zoubir, V. Koivunen, Y. Chakhchoukh, and M. Muma, "Robust Estimation in Signal Processing: A Tutorial-Style Treatment of Fundamental Concepts," *IEEE Signal Processing Magazine*, vol. 29(4), pp. 61–80, 2012.

[7] L. Mihaylova, P. Brasnett, A. Achim, D. Bull, and N. Canagarajah, "Particle filtering with alpha-stable distributions," in *Proceedings of IEEE Workshop on Statistical Signal Processing (SSP)*, 2005.

[8] F. E. Hawary and Y. Jing, "Robust Regression-Based EKF for Tracking Underwater Targets," *IEEE Journal of Oceanic Engineering*, vol. 20(1), pp. 31–41, 1995.

[9] S. J. Godsill, "Bayesian enhancement of speech and audio signals which can be modelled as ARMA processes," *International Statistical Review*, vol. 65(1), pp. 01–21, 1996.

[10] S. Fruhwirth-Schnatter, *Finite Mixture and Markov Switching Models*. Springer, 2007.

[11] R. Adler, R. Feldman, and M. Taqqu, *A Practical Guide to Heavy Tails: Statistical Techniques and Applications*. Springer, 1998.

[12] S. Ross, *Introductory Statistics*. McGraw-Hill,New York, 1996.

[13] R. Pearson, "Outliers in process modeling and identification," *IEEE Trans. Control Syst. Technol.*, vol. 10(1), pp. 55–63, 2002.

[14] G. Agamennoni, J. Nieto, and E. Nebot, "An outlier-robust Kalman filter," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2011.

[15] R. Piché, S. Särkkä, and J. Hartikainen, "Recursive outlier-robust filtering and smoothing for nonlinear systems using the multivariate Student-t distribution," in *Proceedings of IEEE International Workshop On Machine Learning For Signal Processing (MLSP2012)*, 2012.

[16] M. Roth, "Kalman Filters for Nonlinear Systems and Heavy-Tailed Noise," Linköping Studies in Science and Technology. Thesis No. 1613, Linköping University, Sweden, 2013.

[17] J. Ting, E. Theodorou, and S. Schaal, "Learning an outlier-robust Kalman filter," in *Proceedings of European Conference on Machine Learning (ECML)*, 2007.

[18] A. Sinha, T. Kirubarajan, and Y. Bar-Shalom, "Application of the Kalman-Levy Filter for Tracking Maneuvering Targets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 43(3), pp. 1099–1107, 2007.

[19] M. Kok, "Probabilistic modeling for positioning applications using inertial sensors," Linköping Studies in Science and Technology. Thesis No. 1656, Linköping University, Sweden, 2014.

[20] H. Nurminen, T. Ardeshiri, R. Piché, and F. Gustafsson, "Robust Inference for State-Space Models with Skewed Measurement Noise," *arXiv: 1503.06606v1*, 2015.

[21] N. Shephard, *Statistical aspects of ARCH and stochastic volatility*, edited by Cox, D. R. and Hinkley, D. V. and Barndorff-Neilsen, O. E ed. London: Chapman & Hall, 1996, pp. 1–67.

[22] R. Douc, E. Moulines, and D. Stoffer, *Nonlinear Time Series: Theory, Methods and Applications with R Examples*. Chapman & HallCRC, 2014.

[23] E. Ozkan, V. Smidl, S. Saha, C. Lundquist, and F. Gustafsson, "Marginalized adaptive particle filtering for non-linear models with unknown time-varying noise parameters," *Automatica*, vol. 49, no. 6, pp. 1566–1575, 2013.

[24] O. Cappe, E. Moulines, and T. Ryden, *Inference in hidden Markov models*. Springer-Verlag, NY, 2005.

[25] R. Meinhold and N. Singpurwalla, "Robustification of Kalman filter models," *Journal of American Statistical Association*, vol. 84, pp. 479–486, 1989.

[26] F. J. Giron and J. C. Rojano, "Bayesian Kalman Filtering with Elliptically Contoured Errors," *Journal of American Statistical Association*, vol. 81, pp. 390–395, 1994.

[27] C. Masreliez, "Approximate non-Gaussian filtering with linear state and observation relations," *IEEE Transactions on Automatic Control*, vol. 20, pp. 107–110, 1975.

[28] Z. Durovic and B. Kovacevic, "Robust estimation with unknown noise statistics," *IEEE Transactions on Automatic Control*, vol. 44(6), pp. 1292–1296, 1999.

[29] P. Del Moral, *Feynman-Kac formulae : Genealogical and Interacting Particle Systems with Applications*. Springer Verlog New York, 2004.

[30] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson, "Obstacles to high-dimensional particle filtering," *Monthly Weather Review*, vol. 136, no. 12, pp. 4629–4640, 2008.

[31] J. Mattingley and S. Boyd, "Real-time Convex optimization in signal processing," *IEEE Signal Processing Magazine*, vol. 27(3), pp. 50–61, 2010.

[32] R. Chen and J. Liu, "Mixture Kalman filters," *Journal of the Royal Statistical. Society Series B*, vol. 62, no. 3, pp. 493–508, 2000.

[33] M. West, "On scale mixtures of normal distributions," *Biometrika*, vol. 74, no. 3, pp. 646–648, 1987.

[34] S. Boris Choy and J. S. Chan, "Scale mixtures distributions in statistical modelling," *Australian & New Zealand Journal of Statistics*, vol. 50, pp. 135–146, 2008.

[35] C. M. Carvalho, N. G. Polson, and J. G. Scott, "The horseshoe estimator for sparse signals," *Biometrica*, vol. 97, no. 2, pp. 465–480, 2010.

[36] O. Barndorff-Nielsen, J. Kent, and M. Sørensen, “Normal variance-mean mixtures and z-distributions,” *International Statistical Review*, vol. 50, pp. 145–159, 1982.

[37] J. Nakajima and Y. Omori, “Stochastic volatility model with leverage and asymmetrically heavy-tailed error using GH skew Students t-distribution,” *Computational Statistics and Data Analysis*, vol. 56, pp. 3690–3704, 2012.

[38] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[39] A. Doucet and A. M. Johansen, “A tutorial on particle filtering and smoothing: Fifteen years later,” in *Oxford Handbook of Nonlinear Filtering*. Oxford University Press, 2011.

[40] F. Gustafsson, “Particle filter theory and practice with positioning applications,” *IEEE Aerosp. Electron. Syst. Mag.*, vol. 25, no. 7, pp. 53–81, Mar. 2010.

[41] P. M. Djuric, J. H. Kotecha, J. Zhang, Y. Huang, T. Ghirmai, M. F. Bugallo, and J. Míguez, “Particle filtering,” *IEEE Signal Processing Magazine*, vol. 20, no. 5, pp. 19–38, 2003.

[42] N. Chopin, “Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference,” *The Annals of Statistics*, vol. 32, no. 6, pp. 2385–2411, 2007.

[43] T. Schön, F. Gustafsson, and P. J. Nordlund, “Marginalized particle filter for mixed linear/nonlinear state space models,” *IEEE Transaction on Signal Processing*, vol. 53, no. 7, pp. 2279–2289, 2005.

[44] S. Saha, G. Hendeby, and F. Gustafsson, *Mixture Kalman Filters and Beyond*, Upadhyay, S. K. and Dey, D. K. and Singh, U. and Loganathan, A. ed. Chapman & Hall/CRC press, 2015.

[45] S. Saha, E. Ozkan, F. Gustafsson, and V. Smidl, “Marginalized particle filters for Bayesian estimation of Gaussian noise parameters.” in *Proceedings of 13th International Conference on Information Fusion (FUSION)*, 2010.

[46] M. J. Lombardi and S. J. Godsill, “On-line Bayesian estimation of signals in symmetric alpha-stable noise,” *IEEE Transactions on Signal Processing*, vol. 54, no. 2, pp. 775–779, 2006.

[47] J. Vila-Valls, C. Fernández-Prades, P. Closas, and J. F. Rubio, “Bayesian filtering for nonlinear state-space models in symmetric alpha-stable measurement noise,” in *Proceedings of 19th European Signal Processing Conference (EUSIPCO 2011)*, 2011.

[48] M. K. Pitt and S. G. Walker, “Constructing stationary time series models using auxiliary variables with applications,” *Journal of the American Statistical Association*, vol. 100, no. 470, pp. 554–564, 2003.

[49] S. Saha, P. K. Mandal, Y. Boers, H. Driessen, and A. Bagchi, “Gaussian proposal density using moment matching in SMC methods,” *Statistics and Computing*, vol. 19(2), pp. 203–208, 2009.

[50] S. Kim, N. Shephard, and S. Chib, “Stochastic volatility: likelihood inference and comparison with arch models,” *Review of Economic Studies*, vol. 65, pp. 361–393, 1998.

[51] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*. Artech House, 2004.

[52] H. A. P. Blom and Y. Bar-Shalom, “The interacting multiple model algorithm for systems with Markovian switching coefficients,” *IEEE Trans. Autom. Control*, vol. 33, no. 8, pp. 780–783, Aug. 1988.